(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 March 2002 (21.03.2002)

PCT

(10) International Publication Number
WO 02/23328 A2

(51) International Patent Classification[7]: G06F 7/60

(21) International Application Number: PCT/US01/42132

(22) International Filing Date:
12 September 2001 (12.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/232,016 12 September 2000 (12.09.2000) US

(71) Applicant *(for all designated States except US)*: **INFOLIBRIA, INC.** [US/US]; 271 Waverly Oaks Road, Waltham, MA 02452 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **HEDDAYA, Abdelsalam, A.** [EG/US]; 266 Woburn Street, Lexington, MA 02420 (US). **MIRDAD, Sulaiman, A.** [SA/SA]; 43 Ibn Muthfir Street, Riyadh 11431 (SA).

(74) Agents: **THIBODEAU, David, J., Jr.** et al.; Hamilton, Brook, Smith & Reynolds, P.C., Two Militia Drive, Lexington, MA 02421 (US).

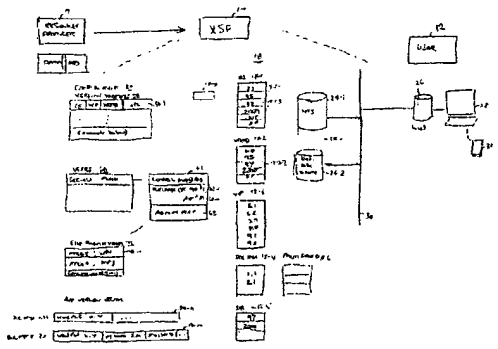
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: MANAGING DISTRIBUTION AND LOCAL EXECUTION OF COMPUTING RESOURCES



(57) Abstract: A configuration management service in a client-server data processing system that manages distribution and installation of different versions and formats of resources so that they remain compatible with each other. When the resources are application programs, the service only distributes combinations of applications that are compatible with one another. This provides the advantages of local execution of application programs on a client computer while at the same time providing the advantages of managed configuration from the central server location. The applications continue to run on the clients and network hosting is not used. The service thus separates execution of applications from the control over their distribution. In a preferred embodiment, a hierarchical, networked cache storage arrangement is used to coordinate the actual physical storage and distribution of the resources as demands are made for them. Profile information maintained for particular users of the service may control how much control over the local file system is given to the service, implementation of different purchase options and different administrative preferences. The invention permits the implementation of a new and cost effective business model for the distribution of software to users that might not necessarily consider it to be cost effective to otherwise make use of such applications, in addition to affording more predictable behavior of installed applications and content.

WO 02/23328 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

MANAGING DISTRIBUTION AND LOCAL EXECUTION OF COMPUTING RESOURCES

RELATED APPLICATION(S)

This application claims the benefit of U.S. Provisional Application No. 60/232,016, filed on September 12, 2000, entitled "Method and Apparatus for Providing Content Banking for Allcache Devices," which is hereby incorporated by reference in its entirety.

BACKGROUND OF THE INVENTION

Architectures for supporting user access to data processing systems have evolved significantly over the past several decades. From a timesharing mainframe computing environment most popular in the late 1960's, there was a profound shift to the use of personal computing (PC) systems during the early 1980's. In the PC model, application programs run on desktop devices that are largely installed and run under the control of individual users.

There are a number of difficulties with pushing application execution to the desktop. In particular, a model where users can simply buy software off the shelf and are left to install and configure such software on their own leads to a host of problems. There is the problem of not only insuring compatibility of specific hardware devices with particular applications, but also with compatibility among the different applications themselves. For example, some applications may not be compatible with certain later versions of operating systems. Likewise, certain applications may themselves not be compatible with the deployment of other applications on the same machine with which they need to communicate. Applications that require the use of special hardware devices, such as graphics, processors, or game controllers, may have compatibility conflicts with the device drivers causing conflicts in peripheral device drivers and the like.

Another list of problems relate to file format compatibility with software applications for viewing or manipulating these files. Examples include video formats, document formats (PDF, Postscript, DOC, compressed archives (e.g., ZIP, etc.).

Because of these limitations in the deployment of applications at the desktop, and for other reasons such as performance, there evolved a client-server type computing model.

With this architecture, applications run on centralized server computers. The servers are connected via networking technologies to client PC computers for data transfer; the PCs thus serve principally as user interfaces.

Most recently the proliferation of internetworks and public networks, exemplified by Internet, have resulted in yet another computing model, which is the so called Application Service Provider (ASP) model. This approach uses central server computers to provide a contractual service offering to deploy, host, manage and, in effect, rent access to software applications from a centrally managed facility.

The client-server and ASP models do alleviate certain difficulties associated with application deployment on the desktop. Providing centralized access to critical applications can be quite cost effective, especially in a business environment, where many users share access to the identical application programs. This model also provides for centralized and uniform deployment of applications which then alleviates problems with version control, remote support, multiple different system configurations and data replication issues. When confronted with thousands of uses in an organization, the cost of application ownership can therefore be effectively controlled by these models for enterprise wide deployment.

However, the situation is still quite difficult for the personal or home computer user as well as for certain types of business users. These users are still left to their own devices in installing software applications, ensuring compatibility among versions, and dealing with any conflicts or other issues which may arise in their installation and execution.

Similarly situated is a business user, for example, located in a small enterprise or in a satellite office of a larger business, who do not have the resources of a corporate information technology professional to oversee installation of their application programs. Other problems are presented to the home user with only a dial-up connection to a network, or a business user who may be on the road often and relying principally on laptop, handheld or other portable computing devices with no or slow-speed wired or wireless network connections. These users do not have access to centralized ASP or client server computing model at all, since their network connections do not support it.

Because they lack access to high bandwidth networks and powerful desktop computers, or a corporate information technology staff, these classes of users are left

presently to their own devices to install desktop applications, and troubleshoot the inevitable conflicts among application versions and vendors.

In addition, it appears as though the standard software distribution model for desktop applications themselves is becoming mature. There seems to be a particular price point in the range of $50 to $200 for most personal computer applications. While a certain subset of the public does have the financial resources to pay for such application software on a per machine license or "shrink wrap" basis, another subset of computer users are positioned at a lower end of the economic scale. They do not have the resources to pay, say $500 to equip a personal computer with the software that they may only need for limited periods of time. Furthermore, such low end users may not need access to all of the advance features that a typical modern software application such as a word processor program provides. Indeed, most PC users probably make use of a very limited subset of features most of the time and advanced features such as for example, graphics editing, equation editing, grammar checking, automatic formatting, and other similar features may only be used a very small part of the time.

Unfortunately, the shrink wrap distribution model forces the software producer to price their packages to be designed for use by the average user. As a result, the market has become somewhat limited for reaching a lower end economic model.

The mass market for application software could potentially reach additional customers if other distribution models were used. But these models should not necessarily have to rely on available or high-speed network connection as is the case in client-server or ASP computing models. These models could also take advantage of functionality built in to the desktop computer itself in the control and optimization of the over distribution and installation of software applications.

SUMMARY OF THE INVENTION

The present invention is directed to a technique for providing managed distribution and configuration of data processing resources. The resources may include applications and data, typically involving different versions of the same application and different file formats for the data. Data may include not only multimedia data files (video, audio, and images) but also documents (books, articles, papers) and user data (spreadsheets and databases). The

technique makes use of a Configuration Manager (CM) process executing in a extended service provider (XSP) location which may, for example, be implemented as one or more network file servers. The configuration manager stores multiple versions of many different types of data processing resources that may be requested to be installed by subscribers to the service. The CM cooperates with an interposer layer installed in a client computer. When a user attempts to execute or access a selected data processing resource, the interposer layer intercepts storage system calls for the asset, directing them to the configuration manager. The configuration manager then directs the request to an appropriate storage location, such as may be in a managed hierarchy of cache storage devices.

In one embodiment, the configuration manager maintains versions vectors for each of many possible but compatible configurations of software applications to be distributed. For example, one version vector may list a particular combination of an operating system version and one or more compatible application software versions, such as a word processor version and a database program version that are compatible with one another, and with the operating system version. Upon a request for a provisioning by a customer, such as when the customer's PC is booted on or initially configured, the configuration manager executes a process to determine a version vector that is most consistent with the customer request, while still meeting version compatibility constraints. For example, the version vector that contains the most recent versions of application programs that are consistent with the user's request will be selected.

The configuration manager typically maintains information concerning each user as a user profile record. The user profile indicates information containing the scope of file system control that the user wishes to delegate to the configuration manager, purchase options that have been selected for particular applications, a list of the applications themselves and other administrative information such as administrative preferences for the particular user. For example, given user may select a purchase option that is on a per machine copy, on a per use basis, or a subscription. The purchase option may also include additional information that for example, select from a number of different available editions of a selected application, wherein each edition enables a certain feature subset for that application.

Similarly, in the context of multimedia content distribution, the service profile for a user may have associated with it a list of allowed multimedia content types that are supported by the applications associated with the version vector assigned to that user. A file format conversion function can assist users with moving data between applications and devices without being obstructed by format incompatibility.

In a preferred embodiment, the configuration manager also has associated with it a mechanism for storage of authoritative copies of all the resources that may be subscribed to by the users. The authoritative resource content may for example include stored copies of all versions and all editions of all supported application programs. Thus, for an operating system such as Microsoft Windows, the resource storage may have associated with it copies of Windows version 3.1, Windows 95, Windows 98, Windows 2000, Windows XP, Windows ME, Windows NT, and so on. The resource storage may also include multiple versions of application software such as for example, Microsoft Word. In such instance, the resource storage may include authoritative copies of Microsoft Word version 6, Word 95, Word 97, Word 2000, Word XP, and so forth. Also, this resource storage may include authoritative copies of all data files generated, purchased, or collected by the user.

The authoritative resource storage mechanism is made available in a distributed fashion through a networked data storage device that can be accessed not only by the configuration manager, but also by processes executing at the client computer. The authoritative resource storage may be bulk file server type storage or logical disk volume type storage, depending upon the particular environment in which the invention is being deployed.

Most commonly, a network file system type model will be chosen such that the resource storage is a bulk file server that can be accessed through bulk network data transfer protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP) and the higher layer File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP) and other well known networking protocols for Internet file transfer.

In other environments, such as local area or metro-area networks, a Network File System (NFS), or a logical disk volume type storage may be utilized for the resource storage.

The configuration manager controls user requests for version updates in a manner which is consistent with their known compatibility. For example, a user may request an upgrade of his word processor to Microsoft Word Version 'XP'. However, he may presently be running a version of an operating systems such as Windows 95 that is incompatible with such a version upgrade. In such an instance, the configuration manager would not be able to find a version vector that matches, and therefore deny the request for the user for an upgrade. Depending upon administrative preferences as selected by the user, the user may be prompted to see if at that time he wishes to pay for the additional fee associated with a version upgrade of the operating system as well. This avoids a situation then where a user is upgrading an application program such as a word processor, without taking into account ramifications that may exist with other existing installed software.

This feature can be used to not only prevent upgrades by users that can become incompatible with existing software but also to prevent the installation of newer versions of software which may not have drivers or compatibility with existing applications. For example, a user may request upgrade from Windows 98 to Windows 2000. However, the user may have installed a database application that operates with a particular peripheral such as a Palm Pilot hand held computer. A device driver for the handheld exists for Windows 98, but no such driver exists for Windows 2000. The user would then be alerted to the possible undesirability of installing the upgraded operating system software.

The configuration manager is responsible for not only maintaining version configuration and edition configuration information for each of the resources requested by each user of the system but also other functions associated with supporting these users on a subscription basis. These functions may include registration, authorization, authentication, and accounting for various services that users select.

In addition, the system provides for distribution of service patches in a managed fashion, for example, should a patch become available for a particular version of an operating system, the service patch can be distributed in an orderly fashion and restricted only those version vectors for which it is known that the patch upgrade will maintain existing compatibility.

In order to support the distribution of resources by the configuration manager, there is also a modification needed for the operating system and file system layers in the client-

side computer. In particular, an interposer layer is associated with the file system at the client-side computer which intercepts all requests made to the local storage device at the client, and which is cognizant of the remote resource distribution service provided by the configuration manager.

The interposer may initially use an auto-discovery protocol at the time of instantiation to determine if the service is available. If the service is detected, the interposer first executes appropriate processes for registration, authentication and administration of the service.

The interposer will then receive stub information for any applications and data that are to be installed from the configuration manager. The stubs themselves are installed as a next step. A particular service mode indicator will be looked at at this point to determine if the user is expecting to run in a connected or disconnected mode. If the disconnected mode is the selected mode, the resources associated with the users' version vector will be downloaded to a local storage cache. The resources required for use in disconnected mode will be labeled as pinned for the particular installed applications. Should there not be sufficient local storage resources to install all of the requested applications, a prioritization process will be executed based on user preferences.

If however, the user is to run in a connected mode, the stub information will be used to set up a local storage as a persistent cache memory. Decisions to store application software of even components of application software will be made on a performance basis. For example, an algorithm such as Least Frequently Used (LFU) can be used to eject application components that are not pinned from the non-authoritative but persistent local cache.

The interposer at the user location is also responsible for maintaining configuration management information peculiar to the particular user, maintaining control information that is necessary to communicate its present state to the configuration manager and for ensuring that the correct file versions are always used that are compatible with existing installed applications, and manages its local cache memory.

The invention therefore provides for the use of distributed cache memory to support a management service that may be used to manage the configuration of the resources such as installed applications and data. The distribution of applications and content in this

manner supports a new marketing model that allows software producers to distribute new versions or changes in their applications to end users in a much more flexible model than is available through the shrinkwrap software model.

Indeed the flexibility associated with Application Service Providers and/or client server computing models that are well known in corporate organizations can now be made available to the users of personal computers and home locations and/or the occasional or satellite located business user that previously only had to rely on shrinkwrap software models.

In addition, such end users can be charged on a per use or level of functionality desired basis without the need to maintain a continuous high speed connection to a network host. In effect, the distribution model separates data from the control over the data in an application service provider model. This provides all of the benefits of high* speed execution on the client computers, while at the same time maintaining the configuration control advantages provided by a client-server or ASP model, without the need for maintaining high speed continuous network connections between a client and a network application host or server.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

Fig. 1 is a system level diagram for implementing a configuration management service according to the present invention.

Fig.2 is a more detailed diagram of a file system network driver and disk controller layers in the user data processor.

Fig. 3 is a flow diagram for a sequence of steps performed by an interposer layer at the client.

Fig. 4 is a flow diagram of steps performed by an interposer layer and a configuration management layer to execute applications.

Fig. 5 is a flow diagram of a sequence of steps performed by the interposer layer and a configuration management layer to insure compatibility of resource content files among different applications.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Fig. 1 is a system level diagram of elements of a distributed data processing system that implements a configuration management service in accordance with the present invention. The configuration management system may be used to manage the distribution of various computing resources, such as computer programs and/or content files, in a coordinated fashion to ensure compatibility among installed resources without intervention by an end user. The system uses a server computing model for installing and configuring resources, but allows the client computer to maintain local control over execution and use of the resources. Access to installed resources may be controlled through a managed cache hierarchy, or simply using a centralized server.

Elements of the configuration management service are principally implemented by one or more server computers 10 located at a service provider. The service provider is referred to herein with the terminology "XSP" to indicate that it may be physically instantiated in a number of different ways. For example, the "XSP" may be a so-called Internet Service Provider (ISP), Network Service Provider (NSP), Application Service Provider (ASP), or located in any other type of distributed computing environment in which a central data processing system or server 10 may communicate messages and exchange data with a number of connected end user or client computers 12.

A number of different resource producers 14 provide copies of applications software, operating systems, content files such as multi-media files and other data processing elements for which the configuration is to be managed by the system. These various elements or more correctly resources 18 are provisioned for in the system. In particular, a configuration manager 20 controls distribution and access to a given resource 18 in a manner such that its distribution to particular end users remains compatible with the distribution of other resources 18 to the same user 12. More specifically, particular versions of resources 18 will not be deployed for particular users 12 unless they are compatible with other resources 18 already allocated to or requested by the user 12.

It should be assumed that in any practical computing environment the resources 18 will encompass a number of different elements such as operating systems, application programs, and even data file formats. For example, a number of different versions of the Microsoft Windows operating system 18-1 may be made available. Different word processing programs such as Microsoft Word 18-2 or Corel Wordperfect 18-3 will typically also be stored as resources 18. A document management application program 18-4 and a database program 18-5 are also associated with a particular example being discussed.

A given resource 18 may have a number of versions. For example, in the case of resource 18-1 such as the Microsoft Windows Operating System, there may be one resource 18-1-1 that is a copy of Windows 3.1, and other resources may be Windows 95, Windows 98, Windows 2000, Windows NT, and Windows XP.

In addition each version of a resource 18 may have associated with it a number of different editions. For example, Windows 98 resource 18-1-3 may have associated with it a number of different editions (not show in Fig.1) correspond to particular instantiations of the resource that have different subsets of features enabled. For example, a stripped down version of a resource such as Microsoft Word 97 18-2-3 may have only the most commonly used word processing functions such as file creation, text entry, copy, paste, and simple text formatting commands.

Another edition of Word 97 18-2-4 may have additional features such as equation editing and graphic drawing features installed.

All of the resources 18 are stored in a distributed file system such that they may be accessed both by the servers 10 at the XSP 10 as well as the remotely located users 12. In the most common embodiment, the file store will be a network file system 24-1 as shown. This can be implemented as a bulk type file server commonly associated with distributed network access such as a web-based file server. The network file server 24-1 participates in and allows access to elements stored within it through the use of commonly available network file protocols such as the File Transfer Protocol (FTP), Hyper Text Transfer Protocol(HTTP), or other well known bulk file transfer protocols.

Alternatively, however, the resources 18 may be stored in a network disk volume 24-2 that may be accessed by both the server at the XSP10 and the user computers 12. For example, this may be network storage device that is addressable as a disk volume in a local

area network or a metro area network type configuration. It should be understood, of course, that certain resources 18 may be resident on the network file server 24 or in the network disk volume 24-2 or some combination thereof. However, again, it should be understood that in most implementations a network file server 24-1 will typically be preferable, especially to support users 12 that have only limited access to the XSP10 through, for example, a dial up connection.

The user 12 have associated with them a central processing unit 28 as well as local disk storage 26. The users 12 are connected through a network infrastructure 30 to the XSP10. The network infrastructure 30 may be any convenient local or wide area network infrastructure connections. However, in the preferred embodiment it should be understood that the network infrastructure 30 is typically a dial up network, a wireless network or other network which provides only limited bandwidth access between the user 12 and the resources 18 located at the XSP . Thus, for example, the network connection 30 may be a dial up connection through a telephone network such that the user 12 need not be always connected to the XSP in order to execute applications.

The network volumes 24 operate as an authoritative, hierarchical, caching system. Thus, for example, the authoritative copies of the elements 18 are stored in the network storage devices associated with the XSP. The local volumes 26 operate as a non-authoritative persistent cache that is synchronized on a best-effort basis with the remote XSP storage 24. However, from time to time non-authoritative and/or temporarily authoritative copies of the elements 18 may be stored in a local disk 826 associated with the user 12 on a cache controlled basis. The nature of the storage mechanism and cooperation between the network volumes 24 and the local volumes 26 will be discussed in greater detail below.

Associated with the XSP is a configuration manager 20 as has been previously described. The configuration manager 20 maintains a number of different data structures in order to support the coordination of a configuration according to the present invention.

First, a set of version vectors 50 are associated with the applications and/or content to be managed. For example, one version vector 50-1 contains a list of compatible applications, in other words, one possible compatible configuration of elements 18 to be managed. The entries in the version vector 50 indicate a particular version of the operating

system 18-1, Microsoft Word Application 18-2 and Corel Word Perfect version 18-3 that are compatible with one another.

There may be a relatively large number of version vectors 50 associated with many different combinations of deployment of resources 18. However, version vectors 50 contain information such that particular combinations of resources 18 will not be associated with one another that are not compatible with each other. This information can be obtained from the software producers 14 themselves upon initial configuration of the version vectors 50. However, other version vectors may be formed in response to the experience of a particular XSP upon the attempt to deploy certain combinations of particular resources 18.

A version vector 50 is selected upon a particular request by a user 12 for participation in a service. For example, user 12 may request that any given access to a copy of the Windows Operating System, a Microsoft Word word processor and a document management program configuration 18-4. The configuration vectors 50 will then be searched for compatible versions of each of the requested resources. In general, the user will be given access to the most recent version of a combination of resources that are compatible with one another. However, if a particular resource 18 is not compatible with later versions of particular other resources 18, then a different version vector will be assigned to the user.

For example, it may be the case that the available version 1.1 of the document management software is not compatible with any version of Microsoft Word but is compatible with a number of different versions of Word Perfect 18-3. The version vector 50-n will be created in such a way that if the user requests the version 1.1 of the document management software they will not be allowed to install any version of Microsoft Word 18-2, and will be asked to install Word Perfect instead. While also being given the opportunity to also upgrade to a version of the document management application 18-4, such as version 2.1, which is compatible with Microsoft Word 18-2. If the upgrade is selected, version vector 50-n will then be associated with the user.

In another instance, a managed resource may be a particular database program 18-5, the user may have deployed an application such as an interface to a peripheral such as a hand held Palm Pilot 31 through which he attempts to access current data stored in the database 18-5. An earlier version 97 of the database 18-5 may have appropriate driver

software for the Palm Pilot 31. However, the user may, upon making a request for service to the XSP, specify that he wishes to have the most recent version 2000 of the database 18-5 which is not compatible with the driver s/w for the Palm Pilot 31. The version vectors 50 will have captured the information that the most recent version of 2000 of the database 18-5 does not have compatible drivers for the Palm Pilot 31. In other words, a Palm Pilot driver 18-6 does not exist that is compatible with the most recent version of the database 18-5. Thus, the version vectors will serve as a mechanism to enforce a configuration whereby even though the user requests the version 2000 of the database 18-5, the version 97 will be used that is compatible with the Palm Pilot drivers 18-6 that he must also install.

The configuration manager 20 also maintains service profile information 60 associated with each user 12. Such service profile information 60 may include control attributes 61, purchase option information 62 - 1, 62-2, ... 62-N for each of the N applications requested by the user, as well as administrative preference information 63.

The control attribute 61 may indicate to what extent the user wishes to give up control over his local file system 26. While this will be understood in more detail shortly, particular users 12 may wish to retain control over installation of certain resources 18. For example, a particular user may wish to control the installation of a word processor 18-2, but would provide the system with control over various other attributes such as the operating system and other applications.

The purchase options 62 indicate for each particular user 12, the manner in which the cost associated with a resource should be administered. For example, one particular application, such as the operating system 18-1 may be purchased by the user on a per machine basis. However, other applications such as the Word Perfect word processing application 18-3 may be such that the user wishes to have them installed but accessible only on a per access basis. Thus, for example, the user may need only to use Word Perfect once in a while but may wish to have access and is willing to pay for access to it on a per use, per day or per hour or other pro rata basis. Information associated with a particular subscription for a particular resource 18, is thus maintained in this purchase option field within the users' service profile 60.

Other information such as administrative preferences 63 may be associated with each user's service profile. Particularly, a given user may want to be more involved with

selection of how the system 12 as a whole operates. As version conflicts and/or options such as the ability to upgrade to new versions or editions of resources become available and/or necessary, the user may prefer to be interrupted and prompted. On the other hand, other users may wish to simply have the system default to a particular function such as installing such requested upgrades as they become available and/or functionality as it is requested.

Finally it should be noted that the configuration manager 20 also maintains different versions of resources such as content file formats and types. For example, one user 12 may have particular older versions of software installed that can support multi-media content files only of an older type such as the Motion Picture Expert Group Version 2 (MPEG-2) type multi-media files and older audio files such as only .WAV type audio files. This information would be indicated with an entry 71-1 associated with the file format vector 70 and/or the user service profile 60.

Other users that have installed more recent versions of an operating system and application software that have the ability to handle later generation content filed including MPEG-4 video files and MP3 type audio files respectively. As will be understood shortly, having this information available in the user service profile 60 and/or in the file format vector can contribute to further compatible and coordinated operation of applications.

Upon instantiation, the configuration manager 20 not only maintains configuration information for its user 12 as has been described above but also performs other functionality associated with any sort of service provided to a number of users in a data processing system. For example, these functions may include user authentication, authorization, and accounting for the services in a manner that is known in the art.

In addition, the XSP distribution model can be used to provide for service patches associated with particular modifications to installed versions. For example, when a software producer 14 releases a service patch 18-P for a particular resource 18, the service patch 18-P can be stored as another resource with its distribution managed among a number of users 12 in the same manner in which the resources 18 themselves are distributed among the users. Version vectors 50 may be created which provide for proper configuration of and distribution of the service patch 18-P to only those users and their configurations that can properly support the patch.

It should be understood that the user 12 may be typically a home based user or personal computer user that has only limited bandwidth access to the XSP such as through a dial up connection through the network 30. However, the invention can be used to advantage even in a business environment, such as where a small business contains only a small number of unsophisticated computer users or in a satellite office of a larger organization that does not have information technology professionals available to administer version distribution and compatibility among applications. Other business users may seldom connect to a corporate enterprise, such as in the case of sales professionals that are highly mobile and typically using laptop computing equipment. These users are also not necessarily easily subjected to configuration control procedures as in the same manner as the home office desk top computer users.

Proper implementation of a configuration management service in this manner however requires also cooperation of the management of content storage at the user side 12 especially within the local storage 26. In particular, processing elements at the user 12 must also ensure that correct versions of resources 18 are made available for the user 12. These elements cooperate with the configuration manager 20 to ensure that configuration management information is maintained, state information is maintained in persistent storage, and otherwise manage access to the resources 18 in a manner which is as transparent as possible to legacy application programs within the user 12.

The operating system, file system, and network drivers do not need to be modified at the client 12, as shown in Fig. 2. In a data processing system 100 instantiated at the user 12 a number of application programs 101-1 and 101-2 are executing in the user memory space. An operating system 102 also contains the typical elements such as a file system 103, network drivers 104 and disk controller 105. The file system 103 and network drivers 104 operate as in the prior art.

The invention adds an interposer layer 150 to the operating system 102 . The interposer layer 150 performs a number of functions as described below. Principally, the interposer layer 150 is cognizant of the configuration management 20 process that is running in the server 10 at the XSP. When the operating system 102 attempts to instantiate an application it first runs a loader program that asks the file system 103 to locate the specific disk blocks in which the application 101 is stored. The interposer layer 150

intercepts these requests to load and run applications 101. The interposer layer 150 then ensures that only a copy of a compatible version of the requested application is provided for running as requested even though such application may not be necessarily be present on the local disk 26 at the time the running is requested.

The interposer layer 150 thus intercepts all requests to the disk controller 105 through the file system 103, and finds the requested block in cooperation with the authoritative storage device provided by the network file system 24. For example, if the requested block is stored at the local disk 26 it is obtained by request to the disk controller 105. However, if the requested block is remote, it is fetched through the network drivers and the network file system 24. In effect the machine is converted to be a machine that can take advantage of the network file system 24 even if it is not initially configured that way. By the expedient of the interposer layer, this interception request occurs transparently for any existing software.

When the interposer layer 150 needs access to an application stored in the network file system 24, the configuration manager 20 then authenticates authorized accounts for the requested resource 18. Once these processes have been completed the interposer layer 150 then permits the processor 100 to actually execute the application 101. It should be understood that the local storage 26 and the network file server 24 cooperate in a manner that is also transparent to the applications 101. For example, they operate as a coordinated hierarchical cache system whereby authoritative copies of the resources 18 are only located at the NFS 24, but also can be pushed out to the local storage at the clients 12.

Fig. 3 is a more detailed flow chart of the sequence of steps performed by the interposer layer 150. Beginning from an initial boot and/or start state 300 the interposer layer 150 first determines, in state 302, if the configuration service is available. If it is not, then processing continues to a state 303 where the legacy and/or locally installed applications are and/or content are handled as in the prior art.

However, should the service distribution mechanism be available in accordance with the present invention, then a state 305 is entered in which the aforementioned authentication, accounting and authorization procedures proceed. Upon successful completion of these procedures between the interposer layer 150 and the configuration manager 20 at the XSP 20, a state 308 is entered in which resource stub information is

received. The stub information for the requested resources 18 will be a minimal indication of where the interposer layer 150 can locate the requested resources 18 at a later time such as one or more file locators for data stored in the network file system 24 associated with the requested one or more resources 18.

The stubs are then installed in a manner so that to the user it appears as though an application, for example, Microsoft Word, has already been installed on their client computer 12, but in effect the copies of the application software are not on the local disk 26 but all stored on the network file server 24 only.

In step 310 a mode determination is made to determine a manner of execution of the service. If the user is expected to remain connected to the network 30 then a state 312 is entered in which the local storage is treated as a pure cache. Thus, any copy of a resource 18 stored in a local disk 26 will be considered to be a cache copy that can be evicted by the interposer to make room for new resources as needed. The corresponding authoritative copy remains within the higher level cache located at the NFS 24.

A state 314 is then entered in which the interposer layer 150 manages downloading of the requested resources 18 in a manner which is consistent with the availability of the network bandwidth and available storage in the local cache memory 26. Decisions about cache can be made based upon desired performance. For example, and algorithm such as Least Frequently Used (LFU) can be used to eject content that has been previously downloaded from the local cache 26. For more information on the cooperation between the network cache and/or network file system elements 24 and local storage 26 reference can be had to a U. S. Patent No. 6,205,481 filed 17 March 1998 entitled "Protocol For Distributing Fresh Content Among Networked Cache Servers," and assigned to the InfoLibria Inc., the assignee of the present invention, which is incorporated herein by reference.

Returning attention to state 310, if the selected mode is such that the user device 12 is not expected to remain connected to the network, then a state 318 is entered in which the requested resources are downloaded to the local memory 26 and pinned in it. As much of the requested resources 18 will be allowed to be downloaded until such time as the local storage 26 becomes full. While this is happening the local copies will be marked as pinned since the network connection will not be made available on a continuous basis. This prevents the interposer from evicting the pinned resources. Should there be less available

local storage space than is required to store all of the requested resources according to the corresponding users version vector 50 than the user may be prompted to prioritize and/or select from the available resources.

For example, if there is not enough room to store both copies of Microsoft Word and Corel WordPerfect at the user's local disk 26 the user may be prompted to select one word processor or another. This selection and/or other prompting of the user can be kept in the administrative preferences section 63 of the user's service profile 60.

Execution of an application 101 that has been installed in this fashion will continue in a manner which is transparent to the user but may involve further downloading or management and coordination of various parts of the resource 18 among the local storage 26 and network file system 24. This may require the interposer layer 150 to access certain elements of a resource 18 through the network drivers 104 as opposed to the file system 103 in certain images when certain requested portions or disk blocks of a resource 18 are not available on the local storage 26 but only at the NFS 24. However, it should be understood that the interposer layer 150 manages this request and coordinates it in a manner in which is transparent to the applications 101 and/or operating system 102 itself.

Fig. 4 is a block diagram of a process executed by the interposer 150 and configuration manager 20. From an initial idle state 500, the interposer receives a request to start an application in state 501. The configuration manager 20 is then queried in state 502 for an appropriate version of the application. The configuration manager in state 520 then checks the compatibility and purchase options associated with the user profile for the requested application. In state 522, a reply is provided to the interposer with the application version and edition information.

In state 504, the interposer then determines if the required version is present in its local cache 26. If so, then the application can be fetched from the local cache. If not, in state 506, the application must be fetched from remote network storage 24. The configuration manager 20 participates in this process as needed in state 524, returning required application components in utilizing elements of the network file system.

Eventually, in state 510, application execution can begin. Should a state 511 be reached in which an application requests the data file, the configuration manager can be queried in state 512 for an appropriate format. This request will be intercepted by the

interposer with a message to the configuration manager in state 526 to determine compatibility of the application version and the available file formats. If they are available on state 514, then they are fetched from the local cache. If not, in state 516 they are fetched from the remote storage.

The invention can also be used to configure and/or manage the configuration of particular application programs and the particular data files that they create. For example, consider a situation in which a user 12 has connected a video capture device that begins storing output of a video camera as a highly compressed MPEG-4 type file. Turning attention to Fig. 5, the executes the video capture program 410 and starts creating a local MPEG-4 file from a video camera input. In state 412 however, the interposer layer 150 will intercept the request to access local storage 26 when the video camera driver first attempts to create the MPEG-4 file, and create a stub for it. The interposer layer 150 in state 414 will then also notify the configuration manager 20 of the creation of the new stub. The configuration manager 20, running in the XSP, in state 450 will check the compatibility of the MPEG-4 format of the new stub with the existing installed applications on the user 12. This can be done by examining the file format vector 70. If the version vector 70 indicates that MPEG-4 is not compatible with versions of multimedia software associated with the particular user 12, a state 452 is entered in which the configuration manager 20 notifies the interposer 150 to create a compatible replacement stub such as an MPEG-2 stub.

In state 418 the user interposer layer 150 receives this request and creates the MPEG-2 stub. The MPEG-2 stub may be created automatically or in a state 420 notify the user if he wishes to have it created. In any event the processing proceeds at both the user interposer layer 150 and at the configuration manager layer 20 so that the following sequence of events occurs. In particular, in state 460 authoritative copies of both the MPEG-4 content as provided by the video capture application are provided to the configuration manager 20 and stored in network file system 24. In addition the configuration manager 20 will execute a conversion process that creates an MPEG-2 version of the MPEG-4 file that is compatible with versions of the installed software resources 18. Thus, without intervention by the user 12, both the original version and a compatible version of the captured video file will be stored. The conversion can also be done on the user's processor.

In state 462 the MPEG-2 converted version will be implicitly labeled as appropriate for applications to be run by this particular user 12. As a result, in later states 430 application programs that are running on the user 12 will be intercepted by the user interposer 150 in such a way that only the appropriate MPEG-2 version for the applications installed on this particular machine, will be accessed.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

CLAIMS

What is claimed is:

1. A method for managing a data processing system comprising the steps of:

maintaining configuration management information at a central service provider location, the configuration management information representing combinations of compatible versions of particular resources to be deployed within the distributed data processing system; and

a distributed networking file server having portions associated with a service provider and other portions associated with a user client side deployment, distribution of resources being made by the file server in a manner such that the users obtain only access to compatible versions of different resources and whereby user access to the compatible versions is coordinated among distributed cache system.

2. A method as in claim 1 wherein the resources are software programs.

3. A method as in claim 1 wherein the resources are application programs.

4. A method as in claim 1 wherein the resources are operating system programs.

5. A method as in claim 1 wherein the resources are data files.

6. A method as in claim 1 wherein configuration management information includes information regarding a type of purchase options for the resource.

7. A method as in claim 5 where the identified purchase options are selected from a group consisting of a per license copy, per usage, and periodic subscription.

8. A method as in claim 1, such that execution of the applications is controlled at the user side of the system whereas distribution of the applications is controlled form a network side.

9. A method as in claim 8 wherein execution of the applications may occur by the user processor accessing resources located at the network file server or local file cache location in a manner which is transparent to the execution of the resources themselves at the user.

10. A system for distribution of computing resources comprising:

(c) a local storage device located at a client processor, for serving computing resources;

(b) an interposer executing on the client processor, for intercepting requests for resources, and for referring such requests to a configuration manager, and receiving instructions for accessing copies of the resources from such resource manager;

(c) a configuration manager, located at a service provider node, for (i) maintaining resource version compatibility vector, (ii) maintaining a use profile of resource selections, (iii) receiving requests for specific resources from an interposer, and (iv) responding with instructions for how to store a resource version compatible with other resource versions used by the client in the local storage device.

11. An apparatus as in claim 10 wherein the resources are application programs.

12. An apparatus as in claim 10 wherein the resources are data.

13. An apparatus as in claim 10 additionally comprising:
a network storage device, for storing a superset of a plurality of versions for each of multiple resources, the network storage device coordinating distribution and selected resource versions to the local storage device.

14. An apparatus as in claim 11 wherein the version compatibility vector specifies compatibility between two application programs.

15. An apparatus as in claim 10 wherein the resource include both application programs and data.

16. An apparatus as in claim 15 wherein the compatibility vector specifies compatibility between a data format and an application that uses it.

17. An apparatus as in claim 16 wherein the interposer additionally intercepts requests for access to data resources and, cooperating with the configuration manager, effects conversion of data resources to a format compatible with selected application.

18. An apparatus as in claim 10 wherein the interposer associates correct versions of resources in the local cache for use while the client processor is disconnected from the configuration manager.

19. An apparatus as in claim 11 wherein the resources are different editions of application programs.

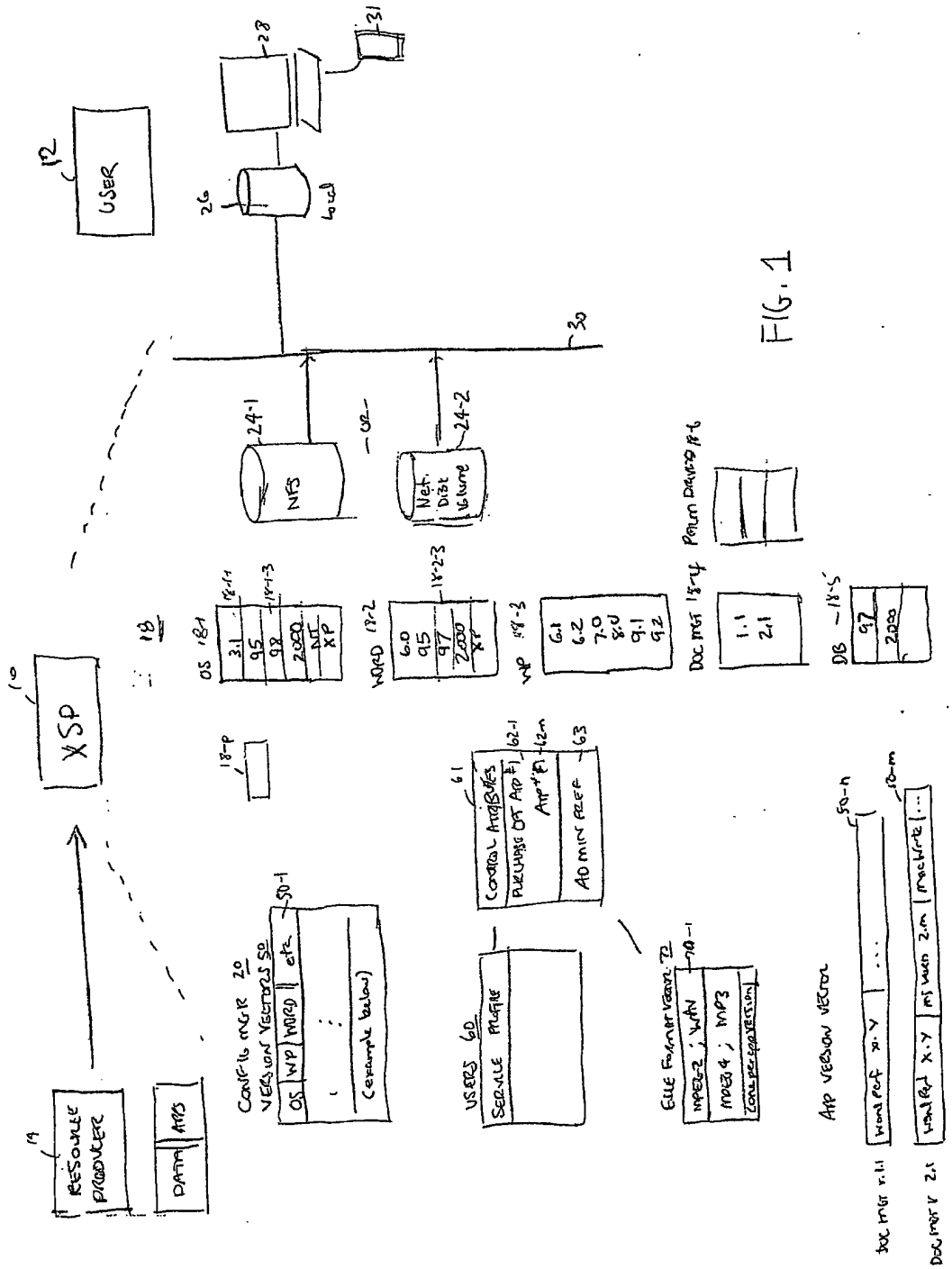

USER
12
28
31
26
Hub
30
24-1
NFS
— OR —
24-2
Net. Disk Volume
XSP
10
18
OS 18-1
3.1
95
98
2000
NT
XP
18-1-1
18-1-3
WORD 18-2
6.0
95
97
2000
XP
18-2-1
18-2-2
18-2-3
VBP 18-3
6.1
6.2
7.0
8.0
9.1
9.2
DOC MGR 18-4
1.1
2.1
PERL DRIVER 18-4
DB 18-5
97
2000
18-P
CONFIG MGR 20
VERSION VECTORS 50
OS
WP
WORD
etc.
50-1
(example below)
61
CONTROL ATTRIBUTES
PURCHASE OF APP #1
ADMIN PERF
62-1
62-m
63
USERS 60
SERVICE PROFILE
ELLE FARMER VBAPP 72
WORD-2 ; WIN
MPEG 4 ; MP3
COMPRESSION VERSION
70-1
APP VERSION VECTOR
50-n
10-m
WORD PERF X.Y
WORD REL X.Y
MS WORD 2.0
MACWRITE
DOC MGR V 1.1
DOC MGR V 2.1

FIG. 1

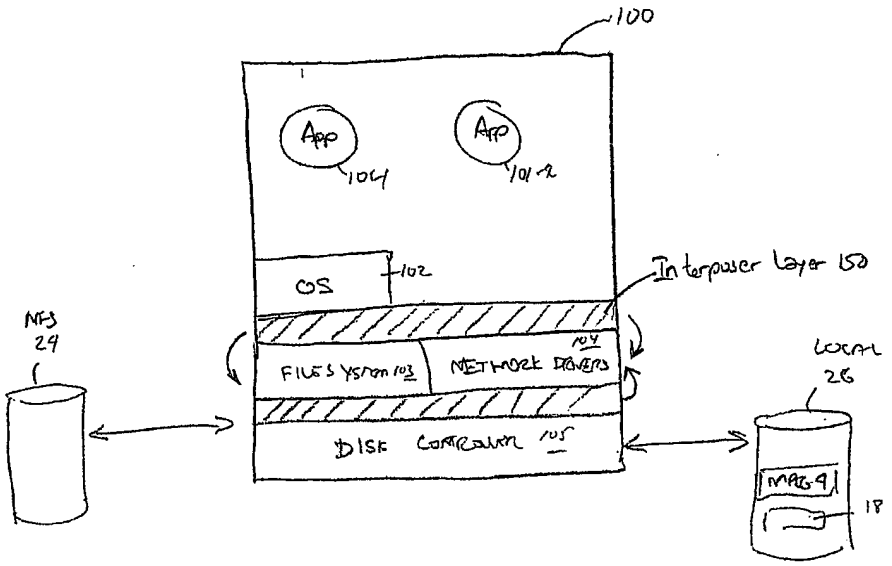

100
App
104
App
104n
OS
102
Interposer Layer 150
NFS
29
FILE SYSTEM 103
NETWORK DRIVERS
104
DISK CONTROLLER 105
LOCAL
26
MAG 9
18

FIG. 2

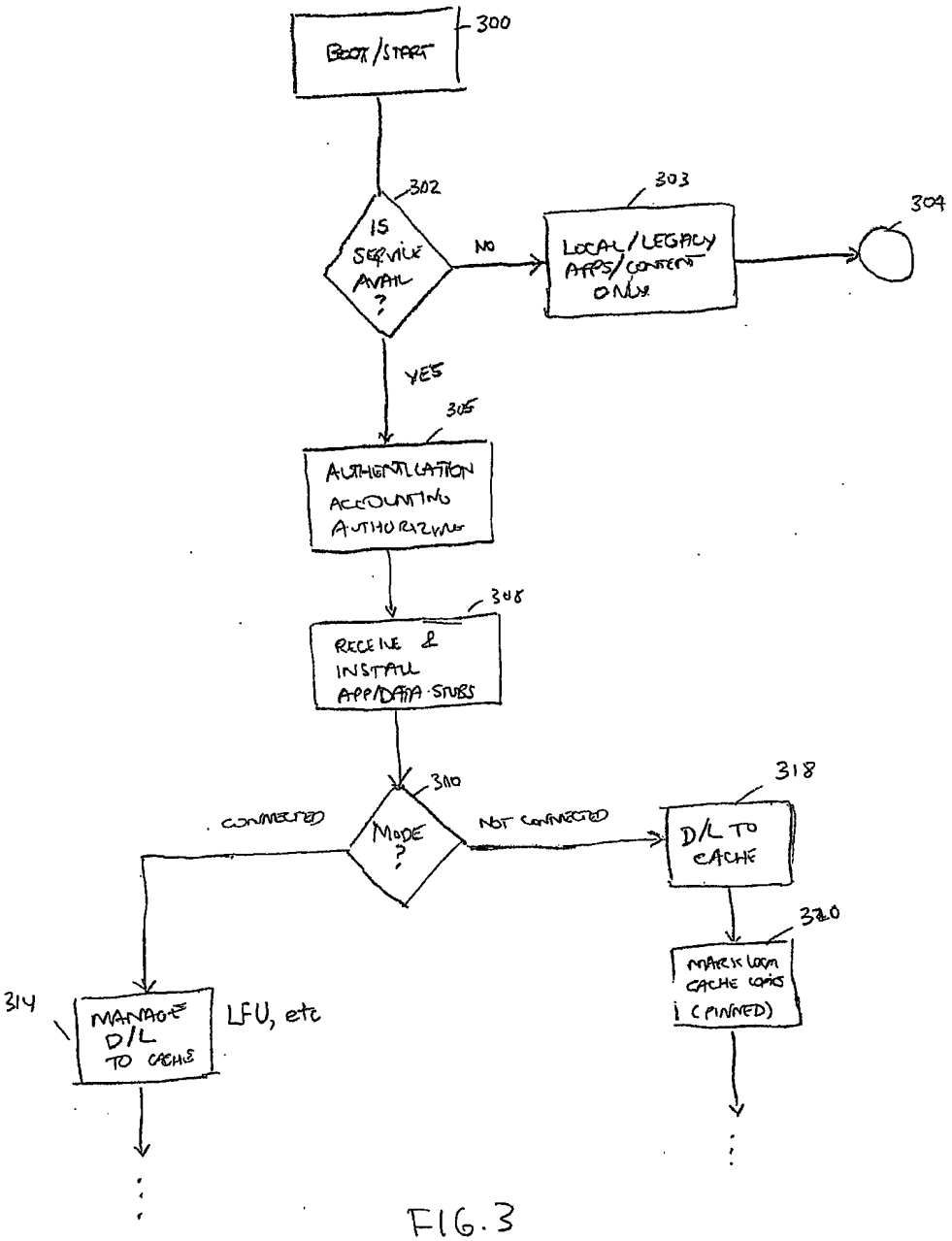
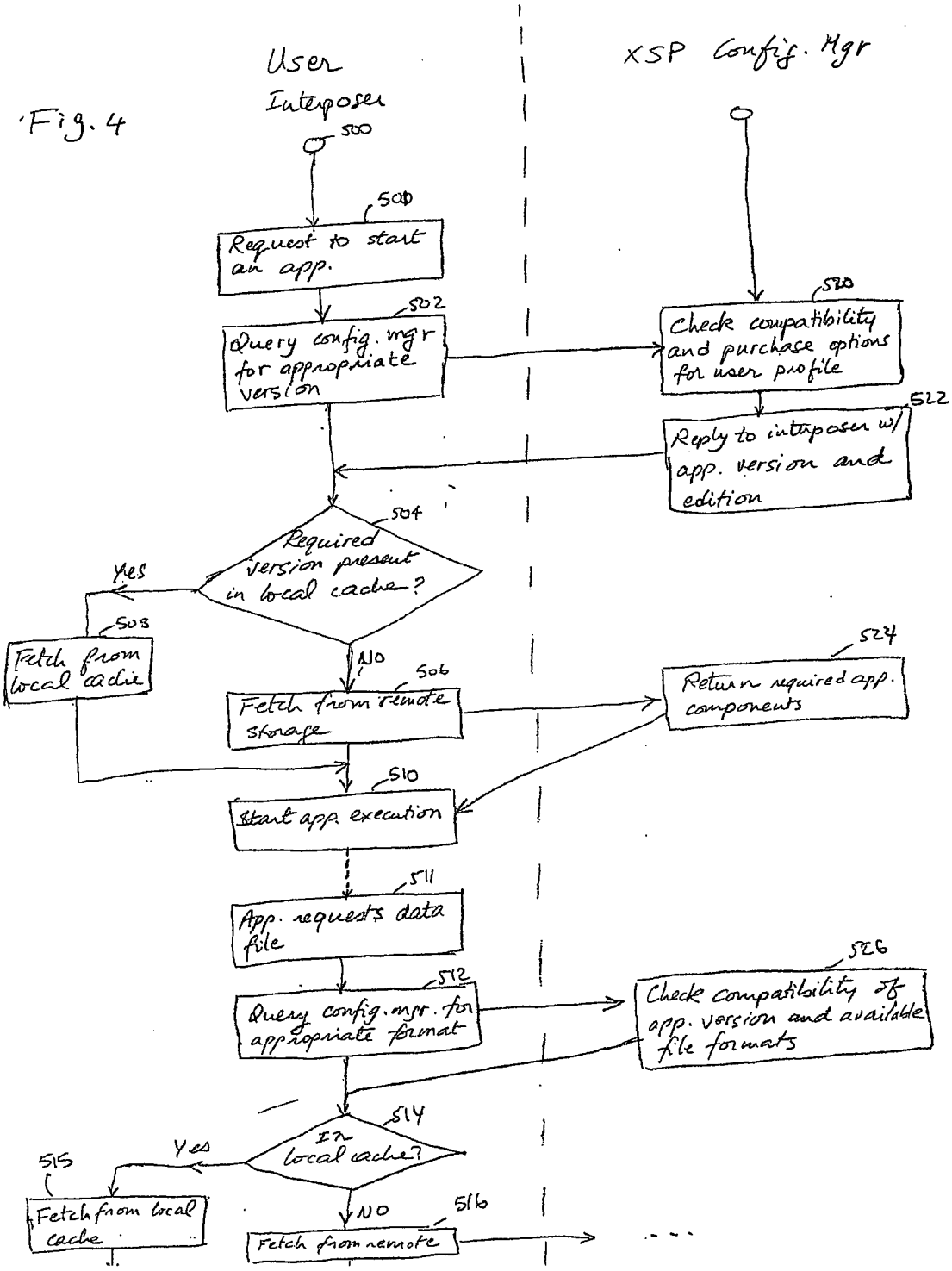

BOOT/START
300
IS SERVICE AVAIL ?
302
NO
LOCAL/LEGACY APPS/CONTENT ONLY
303
304
YES
AUTHENTICATION ACCOUNTING AUTHORIZE
305
RECEIVE & INSTALL APP/DATA STUBS
308
MODE ?
310
CONNECTED
NOT CONNECTED
D/L TO CACHE
318
MARK IN LOCAL CACHE COPIES (PINNED)
320
MANAGE D/L TO CACHE
314
LFU, etc

FIG. 3

Fig. 4


User Interposer
XSP Config. Mgr
500
Request to start an app.
502
Query config. mgr for appropriate version
520
Check compatibility and purchase options for user profile
522
Reply to interposer w/ app. version and edition
504
Required version present in local cache?
Yes
503
Fetch from local cache
NO
506
Fetch from remote storage
524
Return required app. components
510
Start app. execution
511
App. requests data file
512
Query config. mgr. for appropriate format
526
Check compatibility of app. version and available file formats
514
In local cache?
Yes
515
Fetch from local cache
NO
516
Fetch from remote

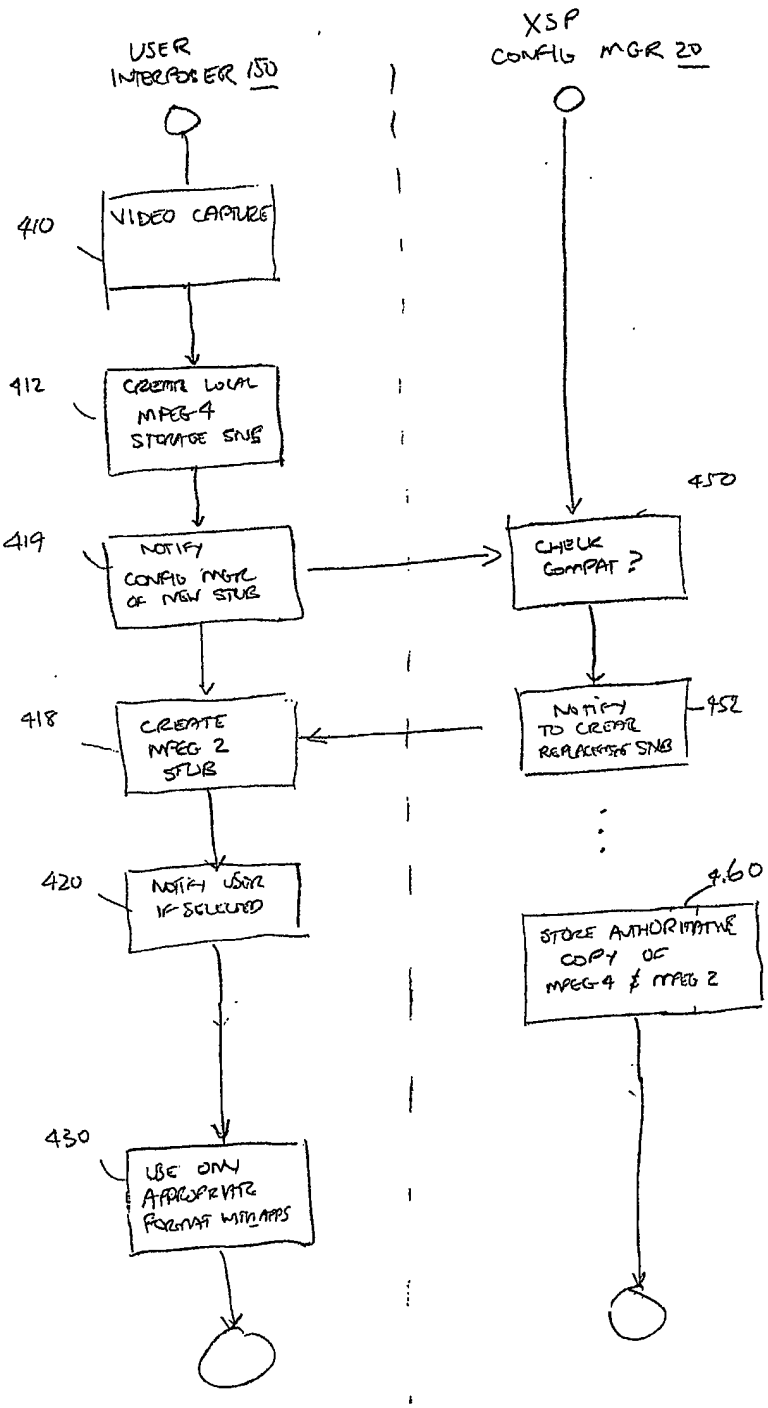

USER INTERFACE 150
XSP CONFIG MGR 20
410
VIDEO CAPTURE
412
CREATE LOCAL MPEG-4 STORAGE SNK
414
NOTIFY CONFIG MGR OF NEW STUB
450
CHECK COMPAT ?
418
CREATE MPEG 2 STUB
452
NOTIFY TO CREATE REPLACEMENT SNK
420
NOTIFY USER IF SECURED
460
STORE AUTHORITATIVE COPY OF MPEG-4 & MPEG 2
430
USE OPTIMAL APPROPRIATE FORMAT WITH APPS

FIG.5

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 March 2002 (21.03.2002)

PCT

(10) International Publication Number
WO 02/023328 A3

(51) International Patent Classification[7]: G06F 9/445

(21) International Application Number: PCT/US01/42132

(22) International Filing Date:
12 September 2001 (12.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/232,016 12 September 2000 (12.09.2000) US

(71) Applicant *(for all designated States except US)*: **INFOLIBRIA, INC.** [US/US]; 271 Waverly Oaks Road, Waltham, MA 02452 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **HEDDAYA, Abdelsalam, A.** [EG/US]; 266 Woburn Street, Lexington, MA 02420 (US). **MIRDAD, Sulaiman, A.** [SA/SA]; 43 Ibn Muthfir Street, Riyadh 11431 (SA).

(74) Agents: **THIBODEAU, David, J., Jr.** et al.; Hamilton, Brook, Smith & Reynolds, P.C., 530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133 (US).

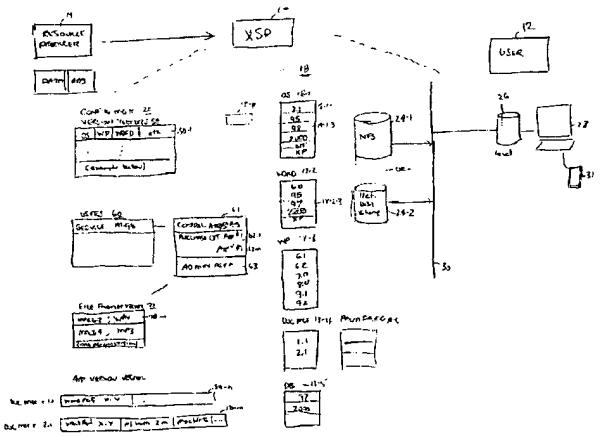
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: MANAGING DISTRIBUTION AND LOCAL EXECUTION OF COMPUTING RESOURCES



(57) Abstract: A configuration management service in a client-server data processing system that manages distribution and installation of different versions and formats of resources so that they remain compatible with each other. When the resources are application programs, the service only distributes combinations of applications that are compatible with one another. This provides the advantages of local execution of application programs on a client computer while at the same time providing the advantages of managed configuration from the central server location. The applications continue to run on the clients and network hosting is not used. The service thus separates execution of applications from the control over their distribution. In a preferred embodiment, a hierarchical, networked cache storage arrangement is used to coordinate the actual physical storage and distribution of the resources as demands are made for them.

WO 02/023328 A3

(88) Date of publication of the international search report:
14 August 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/42132

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/445

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 845 077 A (FAWCETT PHILIP E) 1 December 1998 (1998-12-01) column 2, line 24 -column 3, line 14 column 4, line 66 -column 5, line 60 column 7, line 23 - line 44 ---	1-5,8,9
Y	US 5 499 357 A (SONTY ATASHI C ET AL) 12 March 1996 (1996-03-12) abstract column 5, line 60 -column 7, line 55 ---	1-5,8,9
Y	GB 2 335 768 A (DELL USA LP) 29 September 1999 (1999-09-29) page 3, line 22 -page 4, line 2 page 5, line 7 - line 14 page 9, line 24 -page 10, line 9 page 24, line 19 - line 24 page 30, line 19 -page 31, line 20 --- -/--	10-16,19

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

'A' document defining the general state of the art which is not considered to be of particular relevance

'E' earlier document but published on or after the international filing date

'L' document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

'O' document referring to an oral disclosure, use, exhibition or other means

'P' document published prior to the international filing date but later than the priority date claimed

'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

'X' document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

'Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

'&' document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 April 2003	28/04/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Carciofi, A

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 00 19316 A (INFOLIBRIA INC) 6 April 2000 (2000-04-06)	10-16,19
A	page 6, line 30 -page 7, line 26 page 12, line 28 -page 13, line 21	17,18
A	US 6 094 679 A (JAHROMI BABAK ET AL) 25 July 2000 (2000-07-25) column 3, line 22 -column 4, line 12	1-5
A	US 6 067 582 A (SMITH BENJAMIN HEWITT ET AL) 23 May 2000 (2000-05-23) abstract column 2, line 50 -column 3, line 30 column 9, line 21 - line 50	6,7,10
A	US 5 155 847 A (PORRETT WILLIAM A ET AL) 13 October 1992 (1992-10-13) abstract	9

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interr[illegible]al Application No
PCT/US 01/42132

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5845077 A	01-12-1998	US 6073214 A	06-06-2000
		US 6327617 B1	04-12-2001
		US 2002016956 A1	07-02-2002
US 5499357 A	12-03-1996	BR 9402027 A	13-12-1994
		DE 69428400 D1	31-10-2001
		DE 69428400 T2	14-02-2002
		EP 0632371 A1	04-01-1995
		JP 7006026 A	10-01-1995
GB 2335768 A	29-09-1999	US 6182275 B1	30-01-2001
		AU 8090098 A	12-08-1999
		BR 9803223 A	19-10-1999
		CN 1238494 A	15-12-1999
		DE 19900500 A1	29-07-1999
		FR 2778252 A1	05-11-1999
		IE 980561 A1	28-07-1999
		IT T0990026 A1	26-07-1999
		JP 11213038 A	06-08-1999
		SG 72842 A1	23-05-2000
		TW 403873 B	01-09-2000
WO 0019316 A	06-04-2000	AU 6165299 A	17-04-2000
		EP 1116113 A1	18-07-2001
		WO 0019316 A1	06-04-2000
US 6094679 A	25-07-2000	NONE	
US 6067582 A	23-05-2000	AU 3793997 A	06-03-1998
		EP 0978023 A1	09-02-2000
		WO 9807085 A1	19-02-1998
		US 6532543 B1	11-03-2003
US 5155847 A	13-10-1992	CA 1310131 A1	10-11-1992